

UNIT III – INHERITANCE AND POLYMORPHISM

ABSTRACT CLASSES AND METHODS:

ABSTRACT CLASS:

- An abstract class is a class declared using the keyword `abstract`.
- It may contain abstract methods (methods without a body) as well as concrete methods (methods with implementation).

KEY POINTS:

- Abstract classes cannot be instantiated.
- Abstract classes may contain:
 - ✓ Abstract methods (without implementation).
 - ✓ Non-abstract methods (with implementation).
 - ✓ Constructors, fields, static methods, etc.
- If a class contains at least one abstract method, it must be declared as abstract.
- Subclasses must implement all abstract methods of the abstract class, unless the subclass is also declared abstract.

ABSTRACT METHODS:

- Declared inside an abstract class.
- Do not have a body — only method signature.
- Must be implemented by the subclass.

- **Syntax:**

```
abstract class ClassName
{
    abstract void methodName(); // abstract method
}
```

EXAMPLE PROGRAMS:

```
abstract class Animal
{
    abstract void sound(); // Abstract method (no body)
    void sleep()           // Concrete method
    {
        System.out.println("Animal: Sleeping...");
    }
}
```

```
class Dog extends Animal
{
    void sound()
    {
        System.out.println("Dog: Woof Woof!");
    }
}
class Cat extends Animal
{
    void sound()
    {
        System.out.println("Cat: Meow Meow!");
    }
}
public class TestAbstract
{
    public static void main(String[] args)
    {
        // Animal a = new Animal(); // ✗ Not allowed
        Dog d = new Dog();
        d.sound();
        d.sleep();
        Cat c = new Cat();
        c.sound();
        c.sleep();
    }
}
```

Output:

```
D:\Santhi\Java Programs>javac TestAbstract.java
D:\Santhi\Java Programs>java TestAbstract
Dog: Woof Woof!
Animal: Sleeping...
Cat: Meow Meow!
Animal: Sleeping...
D:\Santhi\Java Programs>
```

```

abstract class Shape
{
    String color;
    Shape(String color)          // Constructor
    {
        this.color = color;
    }
    abstract double area();      // Abstract method
    void displayColor()          // Concrete method
    {
        System.out.println("Color: " + color);
    }
}
class Circle extends Shape
{
    double radius;
    Circle(String color, double radius)
    {
        super(color);
        this.radius = radius;
    }
    double area()
    {
        return Math.PI * radius * radius;
    }
}
public class AbstractExample
{
    public static void main(String[] args)
    {
        Shape s1 = new Circle("Red", 5);
        s1.displayColor();
        System.out.println("Area: " + s1.area());
        Shape s2 = new Circle("Green", 10);
        s2.displayColor();
        System.out.println("Area: " + s2.area());
    }
}

```

Output:

```
D:\Santhi\Java Programs>javac AbstractExample.java
D:\Santhi\Java Programs>java AbstractExample
Color: Red
Area: 78.53981633974483
Color: Green
Area: 314.1592653589793
D:\Santhi\Java Programs>
```